## AMENDMENTS IN THE CLAIMS

1. (currently amended) In a data processing system having a coherent memory hierarchy that includes a memory and a plurality of caches each assigned to particular ones of a plurality of devices that generate cache access operations, a method of maintaining cache coherency comprising:

generating a first-type address operation for a speculatively-issued cache line overwrite operation;

speculatively issuing the first-type address operation for sole ownership of the cache line;

when a first device issues ~~a particular~~ the first-type address operation, which operation requests sole ownership of a cache line and indicates that said first device intends to overwrite the cache line in a first cache, changing a coherency state of the cache line within said first cache to a first coherency state, which indicates that the first device has sole ownership of the cache line AND ~~may or~~ may not overwrite the cache line, wherein said ~~particular~~ first-type address operation further causes a second device that has a most coherent copy of the cache line to not issue the most coherent copy of the cache line on the system bus;

in response to snooping said ~~particular~~ first-type address operation, changing a coherency state of the cache line in a second cache associated with a snooping device to a second state without sending data from said cache line in the second cache to the first cache, wherein a default response to a snoop of a different-type address operation requesting the cache line automatically triggers a return of the cache line from the second cache when the second cache has the most coherent copy of the cache line;

wherein sole ownership of said cache line is provided to said first device without data being sourced to said first cache from another cache; and

determining whether said cache line overwrite operation was correctly speculated, wherein said first coherency state is changed to another coherency state depending on whether said cache line overwrite operation was correctly speculated, and when said operation was correctly speculated:

determining that said cache line overwrite operation was correctly speculated;

initiating a write of said cache line with data provided by said first device; and

changing said first state to a third state indicating that a most coherent copy of said data exists within the cache line of the first cache.

2. (original) The method of Claim 1, wherein, when said first device subsequently initiates a write of said cache line, said method further comprises changing said first state to a third state indicating that a most coherent copy of said data exists within the cache line of the first cache.

3. (canceled)

4. (previously presented) The method of Claim 3, further comprising:

determining when said cache line overwrite operation was not correctly speculated;

changing the coherency state of the cache line in the first cache from said first coherency state to an invalid state; and

subsequently sourcing requests for said cache line from memory.

5. (canceled)

6. (original) The method of Claim 2, further comprising:

snooping requests for access to said cache line at said first cache;

when the cache line in said first cache is in the third coherency state and said first device has completed writing data to said first cache, sourcing the data from the first cache to the second cache;

when the cache line in said first cache is in the invalid coherency state, sourcing the data from memory.

7. (previously presented) The method of Claim 1, further comprising:

snooping requests for access to said cache line at said first cache; and

when the cache line in the first cache is still in the first coherency state, retrying all snooped requests, wherein all subsequent requests snooped while said cache line is in the first coherency state is retried until the coherency state changes.

8. (previously presented) The method of Claim 1, further comprising:

snooping a read request for said cache line at said first cache; and

when the read request receives a null response and said cache line in the first cache is still in the first coherency state:

sourcing data from memory in response to the read request and placing the data in a cache line of a next cache from which the read request was generated; and

changing said first coherency state to an invalid state in said first cache.

9. (previously presented) The method of Claim 1, wherein said first device is an I/O device and said first cache is an I/O cache controlled by an I/O controller, said method further comprising:

issuing the particular address operation as a direct memory access (DMA) Claim from the I/O device in response to a speculative DMA write.

10. (previously presented) The method of Claim 1, wherein said first device is a processor and said first cache is a processor cache controlled by a cache controller, said method further comprising:

issuing the particular address operation in response to a data cache block zero (DCBZ) operation from the processor.

11. (currently amended) In a data processing system having a memory hierarchy that includes a memory and a plurality of caches interconnected by a system bus and each accessible by particular ones of a plurality of devices, a caching mechanism that provides address coherency operations for cache line writes by a first device, said caching mechanism comprising:

a first cache of the first device having a cache line that is a less coherent copy than a corresponding cache line in a second cache of a second device;

a coherency tracking mechanism that supports at least a first coherency state, a second coherency state and a third coherency state, wherein:

said first coherency state indicates that said first cache has sole ownership of the cache line AND data within said first cache line ~~may or~~ may not be overwritten by said

first device, wherein said first cache is provided sole ownership of the cache line pending an outcome of a speculative write operation of the first device that may update the cache line;

said second coherency state indicates that the data is invalid; and

said third coherency state indicates that the data is a most coherent copy of said data;

means for generating said ~~particular~~ first-type address operation for sole ownership of a cache line [[is]] in response to a speculatively-issued cache line overwrite operation processed by the first device;

means for the first device to issue a[[n]] ~~particular~~ first-type address operation requesting sole access to said cache line and indicating that said first device intends to overwrite the cache line in the first cache, wherein said ~~particular~~ first-type address operation further causes a second device that has a most coherent copy of the cache line to not issue the most coherent copy of the cache line on the system bus, said means providing means for speculatively issuing the first-type address operation for sole ownership of the cache line by the first device;

means for changing a coherency state of said cache line within said first cache to said first coherency state when a response is received on said system bus indicating that sole ownership has been granted to said first cache; and

means for determining whether said cache line overwrite operation was correctly speculated, wherein said first coherency state is changed to another coherency state depending on whether said cache line overwrite operation was correctly speculated.

12. (currently amended) The caching mechanism of Claim 11, further comprising:

means for snooping the particular address operation; and

means, when the ~~particular~~ first-type address operation is snooped while the cache line is in said third coherency state within a snooping device's cache and the snooping device determines the ~~particular~~ first-type address operation is for access that does not overwrite the entire cache line, for:

issuing data from the snooping device's cache line on the data bus when said access is granted to said first device; and

changing a coherency state of the snooping device's cache line to a fourth coherency state that indicates that the first device's cache line has data in a coherent state that is as coherent or more coherent than said snooping device's cache line; and

means, when a snooped request of the ~~particular~~ first-type address operation is received while the snooping device's cache line is in said third coherency state and the snooped operation is for sole ownership of the cache line that is to be completely overwritten by the first device, for changing a coherency state of the snooping device's cache line to said second state, and withholding any transfer of data to the first device's cache line, wherein data is not transferred on the system bus from a next cache containing a most coherent copy of the cache line data to the first cache of the first device when said first device indicates it intends to overwriting the cache line and wherein a default response to a snoop of a different-type address operation requesting the cache line automatically triggers a return of the cache line from the second cache to the first cache when the second cache has the most coherent copy of the cache line.

13. (original) The caching mechanism of Claim 12, further comprising:

means for overwriting the data within said first device's cache line with data from the first device, wherein said overwriting is only initiated after sole ownership has been granted and said first cache line is in said first coherency state; and

subsequently changing the coherency state of the first device's cache line to the third state indicating that a most coherent copy of said data exists within the first device's cache line.

14. (previously presented) The caching mechanism of Claim 13, further comprising:

means for snooping requests for access to said cache line from a requesting device;

means, when the cache line is in the third coherency state and said first device has completed writing data to said cache line, for sourcing the data from the cache line to the requesting device; and

means, when the cache line is in the second coherency state, for indicating that said data should be sourced from memory.

15. (previously presented) The caching mechanism of Claim 11, further comprising:

means for snooping requests for access to said cache line from a requesting device; and

means, when the first device's cache line is still in the first coherency state, for retrying all snooped requests, wherein all subsequent requests snooped while said first device's cache line is in the first coherency state are retried until the coherency state changes to a different coherency state.

16. (previously presented) The caching mechanism of Claim 13, further comprising:

means for snooping a read request for said cache line; and

means, when the read request receives a null response and said cache line is still in the first coherency state, for:

indicating that data for the cache line should be sourced from memory; and

changing said first coherency state to the second coherency state.

17. (canceled)

18. (currently amended) The caching mechanism of Claim [[17]] 11, wherein when said cache line overwrite operation was not correctly speculated, said mechanism further comprises:

means for changing the coherency state of the first device's cache line in the first cache from said first coherency state to the second coherency state; and

means for subsequently sourcing requests for said cache line from memory.

19. (currently amended) The caching mechanism of Claim [[17]] 11, wherein when said cache line overwrite operation was correctly speculated, said mechanism further comprises:

means for initiating a write of said first device's cache line with data provided by said first device; and

means for changing the coherency state of the first device's cache line from said first coherency state to the third coherency state.

20. (currently amended) The caching mechanism of Claim 11, wherein said first device is an I/O device and said caching mechanism includes an I/O cache controlled by an I/O controller, and further comprises:

means for issuing the address operation as a direct memory access (DMA) Claim from the I/O device in response to a speculative DMA write.

21. (currently amended) The caching mechanism of Claim 11, wherein said first device is a processor and said caching mechanism includes a processor cache controlled by a cache controller, and further comprises means for issuing the address operation from the processor in response to a data cache block zero (DCBZ) operation.

22. (currently amended) A data processing system, comprising:

an interconnect including an address bus and a data bus;

a plurality of devices interconnected via coupled to said interconnect;

a plurality of caches that are each associated with a device among said plurality of devices, wherein a first cache associated with a first device includes:

a first cache line having a coherency indicator and coherency mechanism that supports at least a first coherency state, a second coherency state and a third coherency state, wherein:

said first coherency state indicates that data within said first cache line is solely owned by the first cache AND ~~may or~~ may not be overwritten by said first device, wherein said first cache is provided sole ownership of the cache line pending an outcome of a speculative write operation by the first device which may update the cache line;

said second coherency state indicates that the data is invalid; and

said third coherency state indicates that the data is a most coherent copy of the data across the plurality of caches;

means for said first device to issue a special address operation that requests sole access to said cache line and indicates that the first device intends to overwrite the cache line in the first cache; and

means for changing a coherency state of said first device's cache line to said first coherency state when a response is received indicating that sole ownership has been granted to said first device.

23. (previously amended) The data processing system of Claim 22, further comprising:

means for a snooping device to snoop the special address operation; and

means, when the snooping device snoops the special address operation while the snooping device's cache line is in said third coherency state and determines that the special address operation is for access that does not overwrite the entire cache line, for:

issuing data from the snooping device's cache line on the data bus when said access is granted to said first device; and

changing a coherency state of the snooping device's cache line to a fourth coherency state that indicates that the first device's cache line has data in a coherent state that is as coherent or more coherent than said snooping device's cache line;

means, when the special address operation is snooped while the snooping device's cache line is in said third coherency state and the snooped operation is for sole ownership of the cache line that is to be completely overwritten by the first device, for changing a coherency state of the snooping device's cache line to said second state, and withholding any transfer of data to the first device's cache line; and

means, when the snooped operation is received, for changing the coherency state of the cache line in the first cache to the first coherency state when sole access is granted to the first device's cache, wherein data is not transferred on the system bus from a next cache containing a most coherent copy of the cache line data to the first cache of the first device when said first device indicates it intends to overwriting the cache line and wherein a default response to a snoop of a different-type address operation requesting the cache line automatically triggers a return of the cache line from the second cache to the first cache when the second cache has the most coherent copy of the cache line.